8提速、可靠、一致

2019年1月5日 22:00

-
- 提高磁盘IO速度的途径

1. 提高文件访问速度的途径：除了改进目录和选好存储结构外就是提快磁盘IO

一. 磁盘高速缓冲(Disk Cache)

1) 磁盘高速缓存：物理上内存中的缓存区，逻辑上是某些盘块的副本

1. 数据交付(Data Delivery)方式
   1) 数据交互：将高速缓冲的数据传送给请求者进程的内存工作区
   2) 指针交互：指向高速缓存的指针交付给请求进程。节省时间
2. 置换算法：类似请求调页/段：LRU，NRU，LFU，具体需要考虑：
   1) 访问频率：高速缓存的访问频率率=磁盘IO频率<快表频率=指令频率
   2) 可预见性：目录块很少再次访问，未满盘口很可能再次访问
      (1) 可预见会再次使用的应放在LRU链末
   3) 数据一致性：修改数据未拷回磁盘会导致不一致
      (1) 应将需要一致性的数据放在LRU链首，优先写回磁盘
3. 周期性写回磁盘
   1) UNIX专设了一个update程序，定期30s左右调用SYNC，强制将高速缓冲中已修改的数据写回磁盘，防止LRU链末损失30s以上工作量

二. 提高磁盘I/O速度的其它方法

1. 提前读(Read-ahead)：顺序访问文件时，读入下一盘块
2. 延迟写：共享资源挂在空闲区链末，到其他进程申请末区时才写回磁盘
3. 优化物理块分布：尽量将文件安排得不分散，减少磁头移动距离
   1) 如采用位示图、以簇为单位都不会分散；链表就容易分散
4. 虚拟盘RAM：用内存空间仿真磁盘，存放obj等临时文件
   1) 虚拟盘的内容由用户控制；高速缓存的内容系统控制

三. 廉价磁盘冗余阵列(RAID，Redundant Array of Inexpensive Disk)

1) 用一台磁盘阵列控制器管理多个相同磁盘驱动器
2) 大幅增加容量、提快IO、增加可靠性

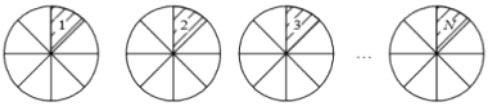
1. 并行交叉存取interleave
   1) 将盘块的数据分成若干子盘块数据，再存储到不同磁盘的同一位置
   2) 并行传输N个磁盘，速度提高N-1倍
   3) 

图 5-29 磁盘并行交叉存取方式

2. RAID 分级
   1) RAID 0。仅提供并行交叉存取。有效地提高磁盘 I/O 速度，但可靠性不好。只要阵列中有一个磁盘损坏，便会造成不可弥补的数据丢失
   2) RAID 1级。有磁盘镜像功能，访问磁盘时，可利用并行读、写特性，将数据分块同时写入主盘和镜像盘。故其比传统的镜像盘速度快，但其磁盘容量的利用率只有 50%，以牺牲磁盘容量为代价
   3) RAID 3级。这是具有并行传输功能的磁盘阵列。它利用一台奇偶校验盘来完成数据校验功能，磁盘的

利用率为 n-1/n。常用于科学计算和图像处理

4) RAID 5级。具有独立传送功能的磁盘阵列。每个驱动器都有独立的数据通路，独立地进行读/写，无专门的校验盘。用来进行纠错的校验信息以螺旋(Spiral)方式散布在所有数据盘上。常用于I/O较频繁的事务处理中
5) RAID 6级。阵列中，设置了一个专用的、可快速访问的异步校验盘。该盘具有独立的数据访问通路，性能改进得很有限，价格昂贵
6) RAID 7级是对RAID 6级的改进，在该阵列中的所有磁盘，都具有较高的传输速率和优异的性能，是目前最高档次的磁盘阵列，但其价格也较高

3. RAID的优点
   1) 可靠性高。除了RAID 0级外都采用了容错技术，可靠性高出了一个数量级
   2) 磁盘I/O速度高。并行交叉存取方式，课提高磁盘数目N-1倍
   3) 性价比高。同体积同容量同速度时，牺牲1/N的容量，3倍速度和3倍便宜

◆

◆ 提高磁盘可靠性的技术

1. 影响数据安全性的因素：人为因素、系统因素、自然因素（详见文件保护）
2. 容错技术fault-tolerant tech：通过在系统中设置冗余部件，提高系统可靠性
3. 磁盘容错技术/系统容错技术SFT：增加冗余的磁盘驱动器、磁盘控制器等

一. 第一级容错技术SFT-I
1. 双份目录和双份文件分配表：在不同磁盘备份FAT
2. 磁盘表面少量缺陷后的补救：
   1) 热修复重定向：取约2%磁盘做热修复重定向区，存放发现缺陷时的数据
   2) 写后读校验：写完一块数据马上读出，送去另一缓冲区，比较原数据，若不同，则重写，再不同，则默认该盘块有缺陷，送去热修复重定向区

二. 第二级容错技术SFT-II
1. 磁盘镜像(Disk Mirroring)：在同一磁盘控制器下增设一完全相同的磁盘驱动器
   1) 每次写数据都要写两遍，磁盘利用率低至50%
2. 磁盘双工(Disk Duplexing)：将两磁盘驱动器再分别接到两个磁盘控制器上
   1) 即两个磁盘都有独立通道，可并行读写

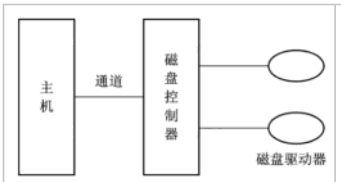


图8-13 磁盘镜像示意图

图8-14 磁盘双工示意图

三. 基于集群技术的容错功能（第三级）
   1) 集群：一组互连的自主计算机组成的计算机系统，人的感觉是一台机器
   2) 即能提高系统并行处理能力，又能提高系统可用性，被广泛使用

1. 双机热备份模式：备份服务器时刻监视着主服务器运行，一旦主服出现故障，备服立刻接替其工作，成为新主服，修复后的原主服会成为新备服
   1) 为连接两台服务器，需要各装一块网卡，建一条镜像服务器链路mirrored server link。如FDDI单模纤允许两服务器距离20公里
   2) 为保证数据同步，需时刻检测主服数据改变，及时用通信系统同步修改到备服的对应数据，为保证通信的高速和安全，一般选高速通信信道
   3) 为保证及时切换，需要配置切换硬件的开关，再备服事先建立好通信配置，迅速处理重新让客户机登录等事宜
   4) 提高了系统可用性，易实现，完全独立，可远程热备份；但备服总是被动等待，系统使用效率仅50%
2. 双机互为备份模式：均为在线服务器，完成不同的任务
   1) 必须先用某专线连接起来，最好再用路由器做备份通信线路
   2) 每台服务器需要两份硬盘，一份装程序，一份接另一台发来的备份数据，即作为另一台的镜像盘
   3) 镜像盘平常对本地用户锁死，保证其数据正确性
   4) 如果通过专线链接检查到了某服务器发生故障，在由路由器验证了故障属实，正常服务器应向连接在故障服务器的客户机广播：需要切换
   5) 而连在非故障服务器上的客户机只会觉得之后网络服务稍慢了些
   6) 故障修复并重新连上网后，服务功能才会被迁回

7) 可扩大到更多台服务器，系统效率很高

图8-15 双机热备份模式

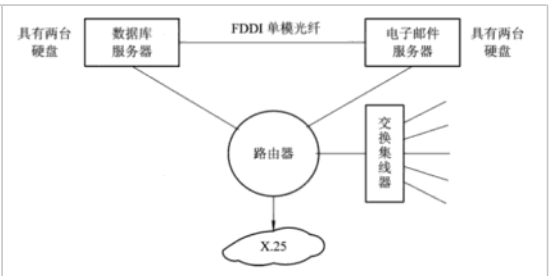


图8-16 双机互为备份系统的示意图

3. 公用磁盘模式：多台计算机连接同一磁盘系统的不同卷
   1) 某计算机故障后，系统调度另一计算机接替，转让卷的所有权
   2) 消除了复制信息的时间，从而减少网络和服务器的开销

四. 后备系统

1) 为防止系统故障和病毒，备份暂时不需要但仍有用的数据和很重要的数据

1. 磁带机：容量大，数十G，便宜；但只能顺序存取，秒速几m

1. 硬盘
   1) 移动磁盘：适用于小型系统、个人电脑。快，脱机方便，保存时间比磁带机稍长，但单位容量贵
   2) 固定硬盘驱动器：适用于大、中型系统。类似双机互备份，每晚将两磁盘的数据互相拷贝到对方的备份区。快，有容错

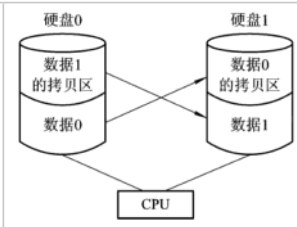


图8-17 利用大容量硬盘兼做后备系统

2. 光盘驱动器
   1) 只读光盘驱动器CD-ROM/DVD-ROM：音视频，不能写，不能后备
   2) 可读写光盘驱动器/刻录机：存数字信息，可读写
      (1) CD-RW刻录机：读写CD、VCD
      (2) COMBO刻录机：读DVD，写CD、VCD
      (3) DVD刻录机：读写CD、VCD、DVD

◆

◆ 数据一致性控制

1. 数据一致性问题：存在不同文件的同一段数据，在任何情况都应相同

一. 事务

1. 事务的定义：访问和修改数据项的程序单位，可视作一系列读写操作
   1) 对不同文件的同一数据读写都结束后才能进行托付操作commit operation/提交操作，结束事务
   2) 任一逻辑错、系统故障导致读写失败后必须进行夭折操作abort operation/回滚操作/取消操作
   3) 夭折事务的数据恢复后，称该事务被退回rolled back
2. 事务的属性（简记为ACID）
   1) Atomic原子性：要么全改，要么不改
   2) Consistent一致性：完成后，所有数据必须一致
   3) Isolated隔离性：并发事务不能在当前事务操作时访问该数据
   4) Durable持久性：事务完成后对系统的影响是永久的
3. 事务记录(Transaction Record)/运行记录log：存储在稳定存储器的数据结构
   1) 包括：事务名、数据项名、修改前旧值、修改后新值
   2) 事务记录表中每行都描述了事务运行时的重要操作：开始、修改、托付、夭折
4. 恢复算法：利用事务记录表处理故障，不致使故障导致非易失性存储器中信息丢失
   1) undo<Ti>：把事务Ti修改过的所有数据恢复为旧值
   2) redo<Ti>：把事务Ti修改过的所有数据设置为新值
   3) 发生故障后，对有开始和托付记录的Ti执行redo；对有开始无托付的Ti执行undo

二. 检查点(Check Points)

1. 检查点的作用
   1) 检查点引入目的：减少发生故障时处理事务记录的视角

2) 隔一段时间将内存中的事务记录保存到稳定存储器、修改过的数据输出到稳定存储器、检查点输出到稳定存储器、执行恢复算法
3) 检查点前托付的Ti不用redo

2. 新的恢复例程算法：查找最近检查点前最后事务Ti，从它开始逐个恢复

三. 并发控制(Concurrent Control)

1) 事务顺序性：各事务修改数据项需按顺序互斥访问
2) 并发控制：实现事务顺序性的技术

1. 利用互斥锁exclusive lock实现顺序性
   1) 每个共享对象设一把互斥锁，访问对象前得先获得其锁，完成后再释放锁
   2) 没获得锁的事务需要等待锁住它的事务释放锁，只能宣布运行失败
2. 利用互斥锁和共享锁shared lock实现顺序性
   1) 若事务只想读对象，发现互斥锁和共享锁都还在，则可获取其共享锁
   2) 若想读，但互斥锁不在，也只能等；若想写，但任一锁不在，也必须等

四. 重复数据的一致性问题

1. 重复文件的一致性
   1) UNIX文件目录项中有ASCII文件名和若干索引结点号，结点数决定重复数
   2) 文件被修改时，必须把其他几个拷贝一起修改
      (1) 查找目录，按索引结点号找到索引，修改该物理位置的数据
      (2) 或建立新的文件拷贝，取代原来的文件拷贝
2. 链接数一致性检查
   1) 共享文件的同一索引结点号可能在目录中多次出现
   2) 需要在索引结点中记录其共享次数count，为0时删除该文件
   3) 可遍历目录，在计数器表中记录索引结点实际出现次数
   4) 若count>实际计数值，则可能永远不删除该文件，浪费空间
   5) 若count<实际计数值，则可能提前删除该文件，造成空索引

i.
ii.
iii.
iv.
v.
vi.
vii.
viii.
ix.
x.
xi.
xii.
xiii.
xiv. ---------我是底线-------